BUD-K-0617
April 21, 1994
Office Automation Systems Engineering Department
Office Automation Semiconductor Sales Division
NEC Corporation

RAB^{2}IT-BRAC

Outline of Functions

<1st Edition>

The information contained in this document is being issued in advance of the production cycle for the device and is subject to change without notice. Use it carefully.

CONTENTS

1 OUTLINE OF BRAC

1.1 System Configuration

Fig. 1 shows an example of system configuration using a BRAC.

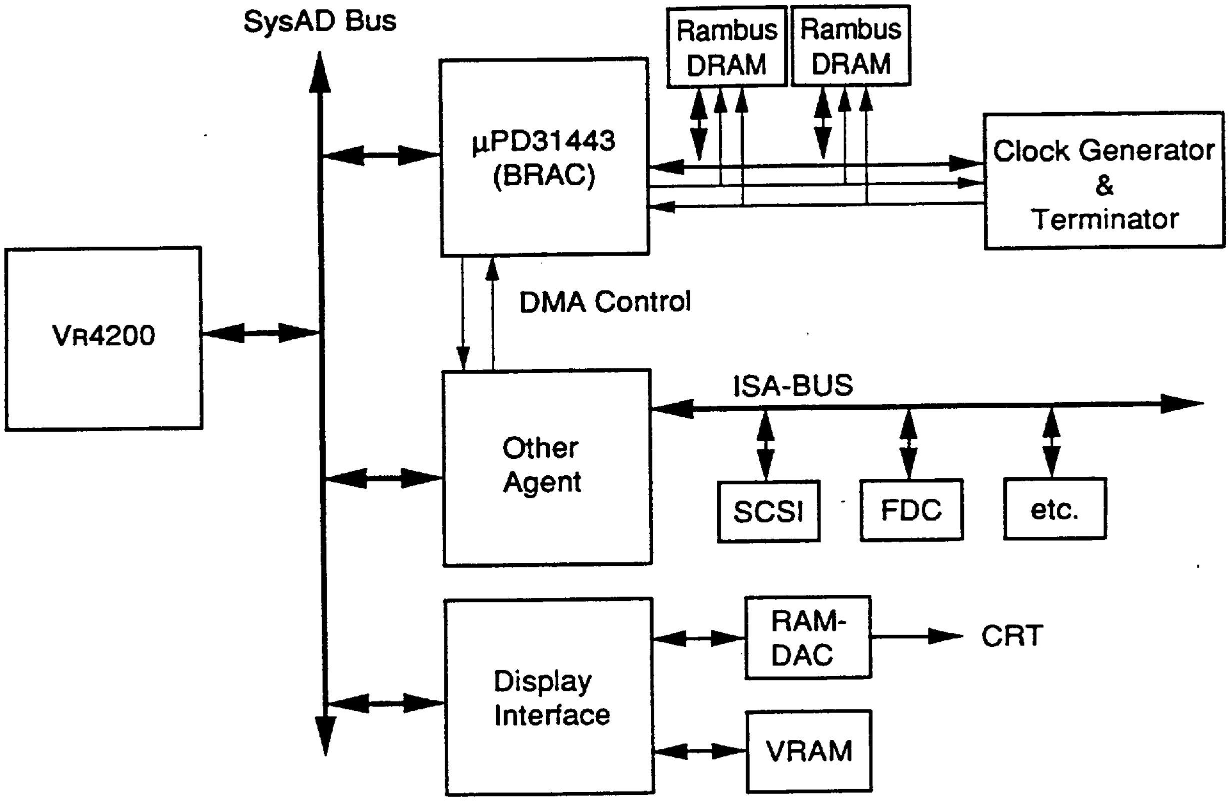


Fig. 1. Example of System Configuration Using BRAC

To configure the system, a VR4200, a BRAC, and several I/O interfaces are required. The BRAC connects RDRAMs as the main memory. The "Other Agent" in the figure performs DMA operations for the BRAC via the SysAD Bus.

1.2 Connection of Rambus Section

Fig. 2 shows a connection example of the Rambus section.

- The termination power supply requires 1.5 A at the maximum.
  In this example, the LT1086CT is used to generate the power.
- The Vref voltage is generated from Vterm by resistance-division.
- A 250-MHz BusClock is generated by the SAW oscillator and buffered by the ECL driver.
- The RDRAM used is the 18-Mbit RDRAM produced by NEC.
  Only this NEC 18-Mbit DRAM (500 MHz, 3.3 V) (PD488170LVN-A50-9) can be connected.
- Connect RDRAMs according to the board wiring pattern specified in the Rambus Technology Guide.
- The termination voltage must be sufficiently stable.

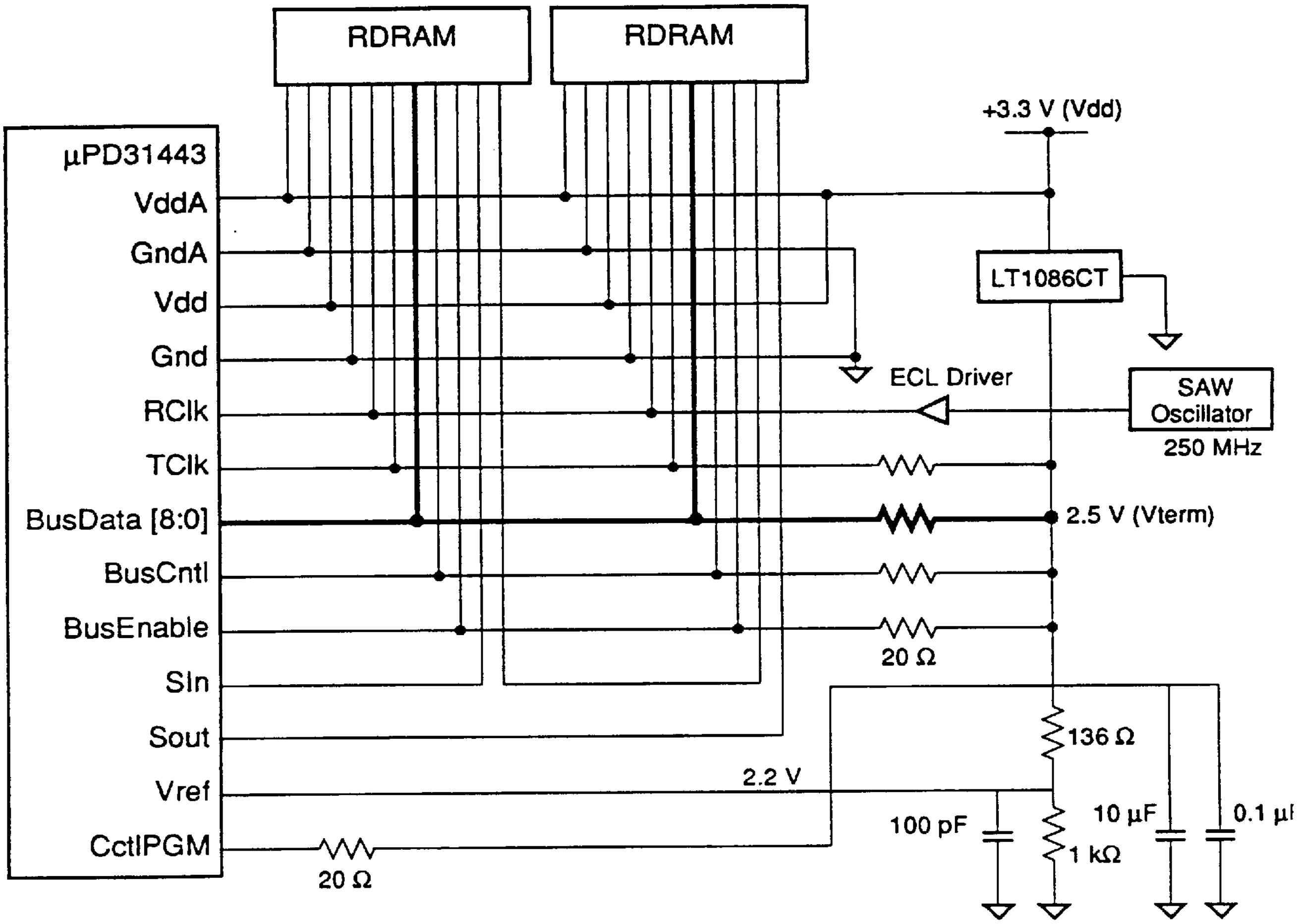


Fig. 2. Connection Example of BRAC Rambus Section

1.3 Address Space

The address space is divided into the Rambus space and BRAC register space. The BRAC register space ranges from 0x1ffff0000 to 0x1ffffffff. The Rambus space is a 36-bit address space. As the VR4200 only has a 33-bit address space, the BRAC uses the lower 33 bits of the Rambus space. In such case, the Rambus address bits [35:33] are all defined as "0".

To initialize the Rambus memory (RDRAM), the BRAC switches its memory space and register space to access them. These two spaces are mapped in the same address area inside the VR4200 space. The register and memory spaces are switched by the BRAC control register.

Fig. 3 shows this switching.

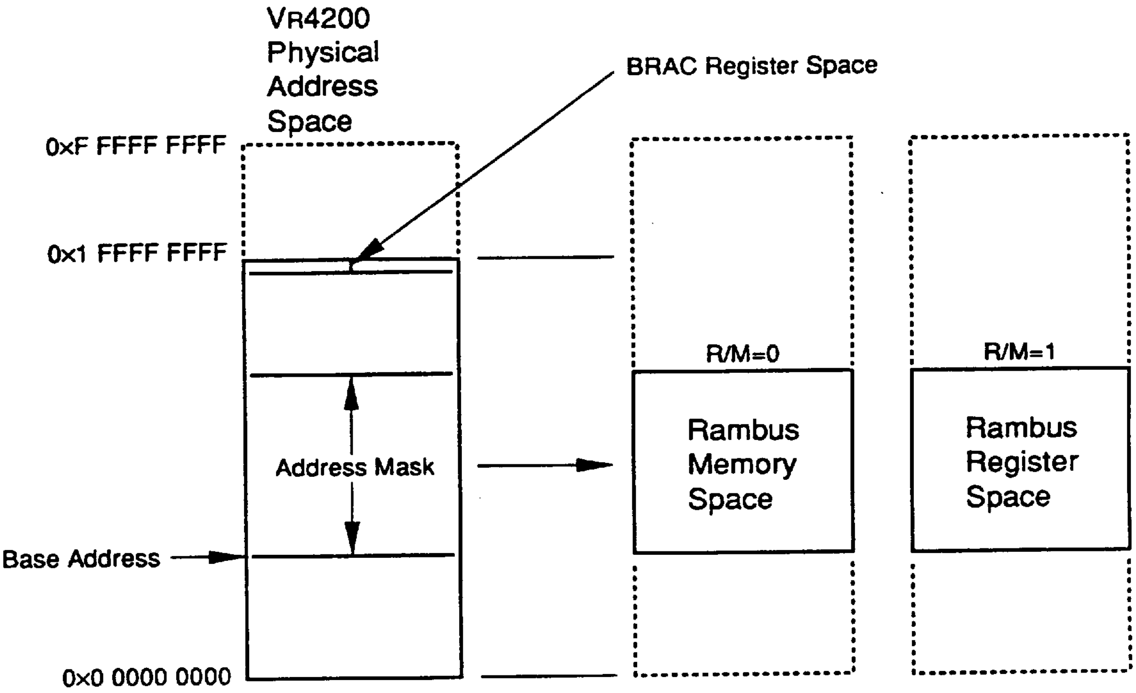


Fig. 3. BRAC Address Space

The Rambus register space must be accessed in order to initialize RDRAMs. The initializing method is described in a later chapter.

As the window to the Rambus space is determined by the base address and mask, a 2n space can be placed at a certain boundary area. The BRAC converts accessing in this window to accessing in the Rambus space.

The VR4200 addresses are used as Rambus space address. The Rambus space is divided into two spaces: register and memory. These spaces are switched and controlled by the mode bit inside the BRAC. Therefore, this means that the register space and memory space of the Rambus are mapped in the same area inside the VR4200 space.

2 BRAC BLOCK CONFIGURATION

Fig. 4 shows the configuration of the BRAC block.

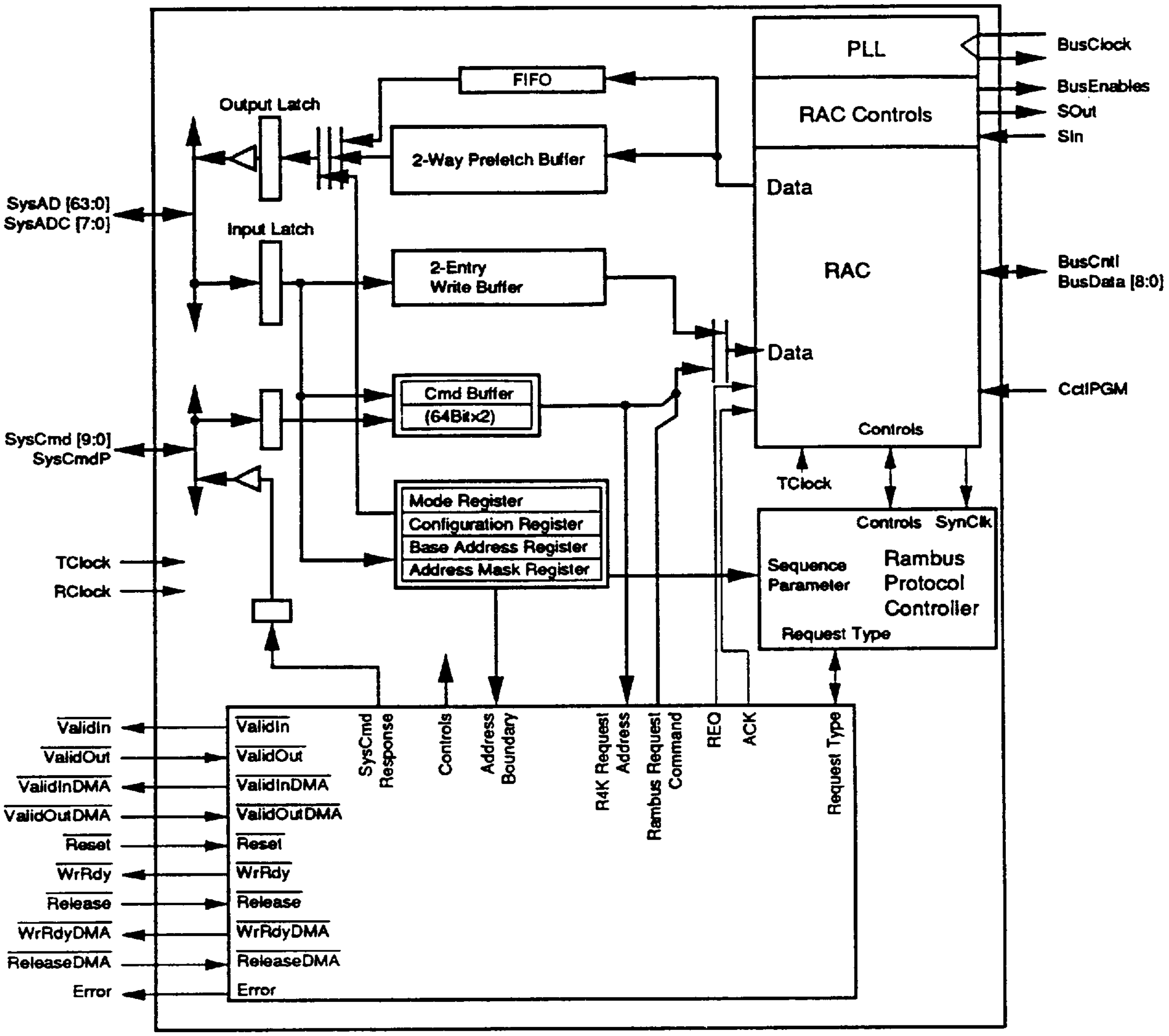


Fig. 4. BRAC Block Diagram

2.1 Internal Configuration

2.1.1 Clocks

The BRAC generates the clock for the Rambus interface using a PLL. It inputs a 250-MHz bus clock from the Rambus, and generates the SynClk divided by four which matches the phase of this bus clock. The RAC (Rambus ASIC Cell) operates in synchronization with SynClk clock.

The VR4200 outputs TClock and RClock based on the operating clock. The VR4200 interface section of the BRAC operates using these two clock inputs from the VR4200.

Fig. 5 shows the distribution of the clocks.

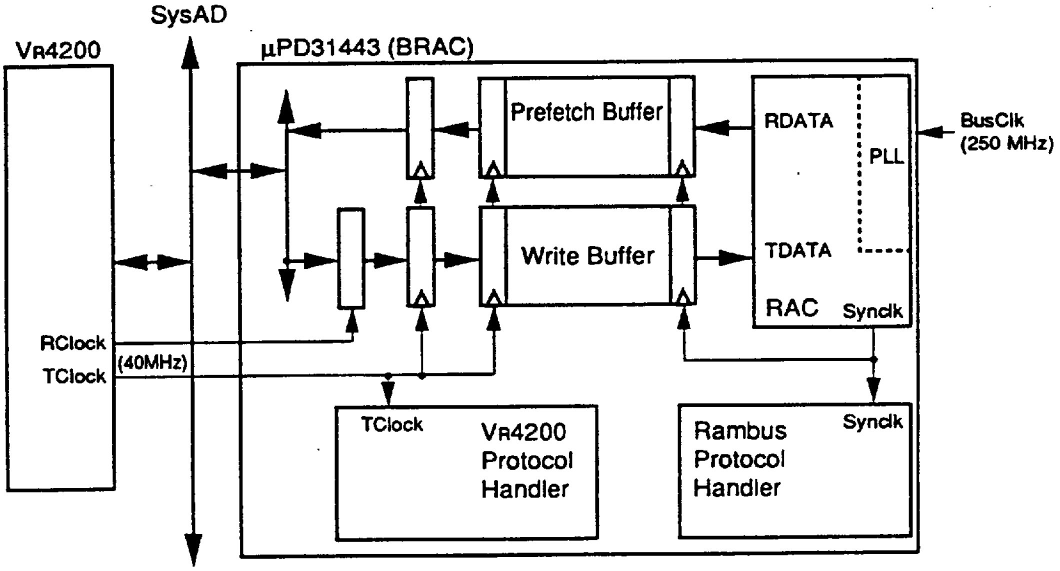


Fig. 5. Distribution of BRAC Clocks

2.1.2 Prefetch buffer/Write buffer

The Prefetch buffer is provided to send responses efficiently to the read requests from the VR4200. The BRAC reads the data of the next address (address read + 64 bytes) beforehand to improve the hit rate.

The write buffer uses the FIFO method to receive data from the VR4200 via the SysAD bus so that the data is not lost.

2.1.3 VR4200 protocol handler

The VR4200 protocol handler is a sequencer which interprets the SysAD protocol of the VR4200 and initiates each unit. It contains the BRAC control register.

2.1.4 Rambus protocol handler

The Rambus protocol handler is a sequencer which supports the protocol for reading and writing the data collected. It receives requests for data transfer from the VR4200 protocol sequencer and controls data transfer between the write buffer/prefetch buffer and the Rambus section.

3 PINS/INPUTS/OUTPUTS

The BRAC has input and output signals of the VR4200 SysAD bus interface and Rambus interface. The following describes the pin signals.

3.1 Input/Output Signals

Table 2 shows the input/output signals of the VR4200 system interface and **Table 1** shows those of the Rambus™ interface.

The logic level of each signal is described in the input/output direction column.

The parentheses in the direction column of **Tables 1** and **2** indicate the following.

The "T" in the input/output direction of **Table 2** means tristate.

(R): Rambus level
(C): CMOS level (Vdd=3.3 V, CMOS Level)
(P): Power supply

Table 1. Rambus Interface Input/Output Signals

Signal	Bit Width	Direction	Description
ClockToMaster	1	I (R)	250-MHz clock input. The Rambus interface operates according to this clock.
ClockFromMaster	1	O (R)	Connected to the ClockToMaster input inside the BRAC.
$\overline{\text{BusCtrl}}$	1	I/O (R)	Signal which transmits the beginning of a request packet, or transmits command packet and acknowledge packet.
BusData<8:0>	9	I/O (R)	Bus which transmits the request packet itself and read/write data packet.
$\overline{\text{BusEnable}}$	1	O (R)	Signals which activates the Rambus channel
SOut	1	O (C)	Output of Daisy-chain which initializes the Rambus device
SIn	1	I (C)	Input of Daisy-chain which initializes the Rambus device
Vref	1	I (*1)	Rambus reference voltage
CctlPGM	1	I (*2)	Drive capability control pin
GndA	1	I (P)	RAC analog ground
VddA	1	I (P)	RAC analog +3.3 V power supply
GndR	1	I (P)	RAC digital ground
VddR	1	I (P)	RAC digital +3.3 V power supply

(*1) Threshold voltage used for determining the logic value of Rambus low-swing signals.

(*2) For controlling current. Connected to the current control resistor connected to Vterm.

Table 2. VR4200 Interface Input/Output Signals 1/2

Signal	Bit Width	Direction	Description
SysAD [63:0]	64	T/I/O (C)	64-bit address/data bus connected to the SysAD of VR4200
SysADC [7:0]	8	T/I/O (C)	8-bit bus including check bits connected to the SysADC of VR4200
SysCmd [8:0]	9	T/I/O (C)	9-bit bus for transmitting the command/data ID connected to the SysCmd of VR4200
SysCmdP	1	T/I/O (C)	1-bit even parity bit for the SysCmd bus connected to the SysCmdP of VR4200
$\overline{\text{ValidIn}}$	1	T/O (C)	Connected to $\overline{\text{ValidIn}}$ of VR4200. Signal which indicates that BRAC is driving valid addresses/data on the SysAD bus and valid commands/data on the SysCmd bus.
$\overline{\text{ValidOut}}$	1	I (C)	Connected to $\overline{\text{ValidOut}}$ of VR4200. Signal which indicates that VR4200 is driving valid addresses/data on the SysAD bus and valid commands/data on the SysCmd bus.
$\overline{\text{ValidInDMA}}$	1	O (C)	Signal for DMA. Same as $\overline{\text{ValidIn}}$. Responds to commands output by the $\overline{\text{ValidOutDMA}}$ signal which indicates that BRAC is driving valid addresses/data on the SysAD bus and valid commands/data on the SysCmd bus.
$\overline{\text{ValidOutDMA}}$	1	I (C)	Signal for DMA. Same as $\overline{\text{ValidOut}}$. Signal which indicates that DMA is driving valid addresses/data on the SysAD bus and valid commands/data on the SysCmd bus.
$\overline{\text{WrRdy}}$	1	T/O (C)	Connected to the $\overline{\text{WrRdy}}$ of VR4200. Signal which informs VR4200 that BRAC can accept write requests.
$\overline{\text{WrRdyDMA}}$		I (C)	Signal for DMA. Same as $\overline{\text{WrRdy}}$. Signal which informs VR4200 that BRAC can accept write requests.
$\overline{\text{Release}}$		I (C)	Connected to the VR4200 $\overline{\text{Release}}$. Signal which indicates that VR4200 will release the system interface to the slave.
$\overline{\text{ExtRqst}}$		I (C)	Connected to the $\overline{\text{ExtRqst}}$ of VR4200. Signal which indicates that other external agents are requesting the bus.

Table 2. VR4200 Interface Input/Output Signals 2/2 (Cont)

Signal	Bit Width	Direction	Description
RClock		I (C)	Connected to the VR4200 RClock. System interface reception clock
TClock		I (C)	Connected to the VR4200 TClock. System interface transmission clock
$\overline{\text{Reset}}$		I (C)	Connected to the VR4200 $\overline{\text{ColdReset}}$. Must be asserted when the power is turned on and when initializing the system.
$\overline{\text{Error}}$		O (C)	Asserted when errors occur while accessing Rambus.
I.C.G.		I (C)	Internally connected. Connect this to Gnd.
I.C.O.		I (C)	Internally connected. Leave this unconnected.
Vdd		(P)	Power supply pin
Gnd		(P)	GND pin

3.2 Pin Configuration

Fig. 6 shows the pin configuration of the BRAC viewed from the top.

Fig. 6. BRAC Pin Configuration (Top View)

3.3 Package

The BRAC is available in the 208-pin plastic QFP with 0.5-mm lead pitch. The body size is 28 mm x 28 mm and the body thickness is 3.2 mm.

4 FUNCTIONS

4.1 Basic Operations

The BRAC has one Rambus interface and one VR4200 system interface (SysAD bus).

The BRAC serves as a slave device for the VR4200 system interface, and is unable to issue its own requests. Therefore, when configuring a VR4200 system using the BRAC, I/O agents, interruption agents, etc., must also be connected to the SysAD bus according to the system requirements.

Other agents for outputting data transfer requests to the BRAC can be used instead of the VR4200. DMA transfer via SysAD is also possible. However, the cache coherency function is not supported on the SysAD.

The Rambus interface of the BRAC is "Non Responding Master" and does not correspond to the Rambus multi-master protocol.

The BRAC arbitrates the RDRAM and agents on the SysAD by interpreting the requests transmitted to the VR4200 system interface (SysAD bus) and transmiting required requests to the Rambus .

Because the BRAC has a prefetch buffer and RDRAMs have sense-amplifier caches, the following three accessing patterns are available for reading.

1. Prefetch buffer hit
2. Prefetch buffer miss, RDRAM hit
3. Prefetch buffer miss, RDRAM miss

During writing, there are also two accessing patterns due to the sense-amplifier cache function of the RDRAM. In addition, when the addresses prefetched by the BRAC receive write requests, the following patterns are also available.

1. RDRAM hit
2. RDRAM miss
3. Invalidation

4.2 VR4200 Interface

This section describes the functions related to the VR4200 interface of the BRAC.
The following shows the system interface requests that the BRAC acknowledges.

[Acknowledged Requests]

- Read (Read request command)
- Write (Write request command)

[Ignored Requests]

- Null (Null request)
- SINull (System interface release Null request command)
- SCNull (Secondary cache release Null request command)
- Ivd (Invalidation request command)
- Upd (Update request command)
- Ivtn (Intervention request command)
- Snoop (Snoop request command)
- RwWF (Read with write forthcoming request command)

All other requests are also ignored (Multi-processor related protocols are not supported.). Transmitting commands requiring responses causes the system to hang-up.

4.3 SysAD Bus DMA

The DMA system can be constructed on the SysAD with the connections shown in **Fig. 7**.

The following describes the DMA mechanism assumed by the BRAC according to the figure below.

The BRAC has two sets of ValidIn/Out. One is for VR4200 and the other is for the DMA master. DMA transfer on the SysAD bus is performed in the following order.

1. SysAD bus mastership is passed to the DMAC by asserting $\overline{\text{ExtRqst}}$.
2. Data is DMA-transferred.
3. "Null" request is issued to the VR4200 for releasing the bus mastership.

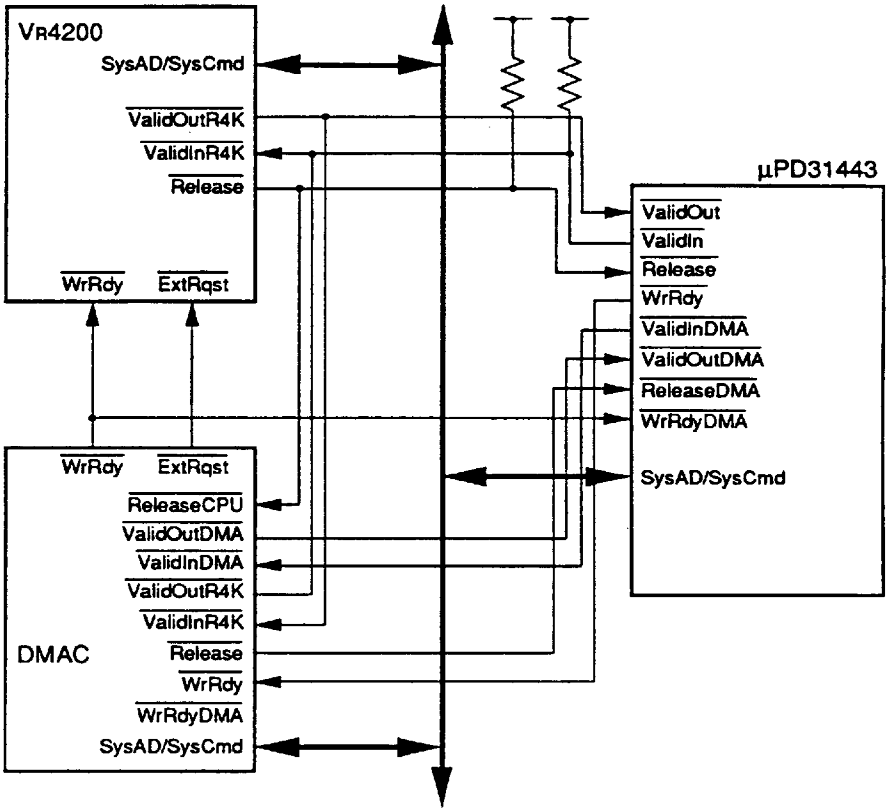


Fig. 7. DMA System on SysAD Bus

The following flowchart shows the DMA operation on the SysAD bus. The assertion of the $\overline{\text{Release}}$ signal after the READ request command of the VR4200 is treated as a bus release signal corresponding to the command output.

For this reason, this system must be designed so that, to perform DMA transfers, the DMA master acquires the SysAD bus mastership only when there are no pending commands on the bus.

After the command is executed, the DMA master issues a Null request command to the VR4200 to release the bus mastership.

The DMA master must also be designed so that no more than two read requests or write requests are issued to the BRAC.

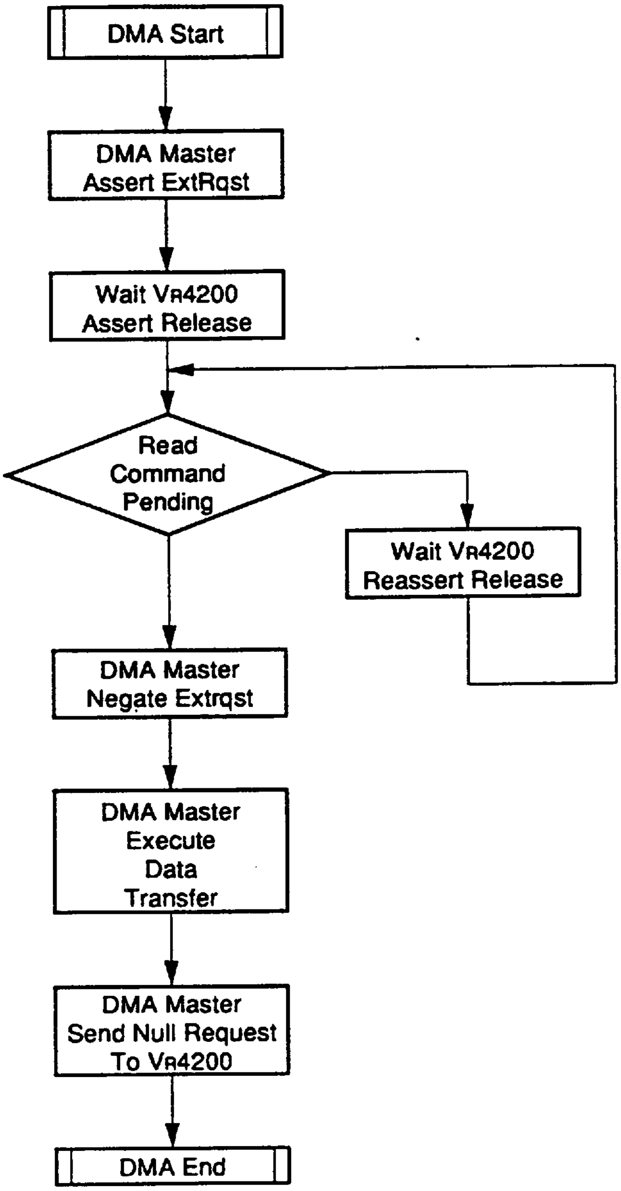

DMA Start
DMA Master Assert ExtRqst
Wait VR4200 Assert Release
Read Command Pending
Wait VR4200 Reassert Release
DMA Master Negate Extrqst
DMA Master Execute Data Transfer
DMA Master Send Null Request To VR4200
DMA End

Fig. 8. DMA Operation on SysAD Bus

4.4 SysAD Bus Protocol

4.4.1 Read request

Fig. 9 shows read request packets supported by the BRAC.

SysCmd	8	7	6	5	4	3	2	1	0
Block Read	0.	0	0	0	00/01/10		Link	Block Size	
Single Read	0	0	0	0	1	1	Read Size		

Fig. 9. Read Request Packets BRAC Acknowledges

[Block Read]

The block read command reads data consisting of 16 bytes or more as one operation, and returns an error-free response data. It requires coherent/exclusivity. For coherent block read/ non-coherent block read attributes, the same BRAC operations are performed.

When the block read command is being executed, the lower bits of the accessed address indicate the sub-block ordering address inside the boundary shown in **Table 3** according to the block size.

- Link : This field is ignored.
- Block size : Block transfer is executed according to the number of this field as shown in **Table 3**.

Table 3. Block Transfer Size

SysCmd [1:0]	Transfer Size	Boundary
0	64 bits x 2	16 bytes
1	64 bits x 4	32 bytes

[Single Read]

The single read command reads data consisting of 8 bytes or less, and returns error-free response data. The value read from an RDRAM is returned as is for parity check.

- Data consisting of 8 bytes or less are all read in 8-byte units.
  This is because the VR4200 performs parity check for all bits of the response packet even when accessing data below 8 bytes. When the 8-byte data including the data corresponding to the memory read request has an error, and even if the error is not located in the data included in the desired address, the parity error is generated.

4.4.2 Sub block ordering

Sub block ordering is the order of reading the data block when the data transferred first is not at the top of the data block.

The BRAC transmits the response data according to the sub block ordering rule of the VR4200. In this process, the binary counter output which counts the double-words inside the block and the start address are exclusive-ORed bit by bit.

4.4.3 Write request

Fig. 10 shows the write request packets supported by BRAC.

SysCmd	8	7	6	5	4	3	2	1	0
Block Write	0	0	1	0	1	0	Link	Block Size	
Single Write	0	0	1	0	1	1	Write Size		
Write Error	0	0	1	0	0	x	x	x	x

Fig. 10. Write Request Packets Accepted by BRAC

[Block Write]

The block write command writes data consisting of 16 bytes or more in one operation.

- Link : This field is ignored.
- Block size : Write block transfer is executed according to the number of this field as shown in **Table 4**.

Table 4. Block Transfer Size

SysCmd [1:0]	Transfer Size	Boundary
0	64 bits x 2	16 bytes
1	64 bits x 4	32 bytes

[Single Write]

The single write command writes data consisting of 8 bytes or less.

- 8-byte data (64 bits) to be written can be accessed in 1-byte units.
  Table 5 shows the valid transfer sizes.

Table 5. Write Data Sizes

SysCmd [2:0]	Transfer Size
0	1 byte valid
1	2 bytes valid
2	3 bytes valid
3	4 bytes valid
4	5 bytes valid
5	6 bytes valid
6	7 bytes valid
7	8 bytes valid

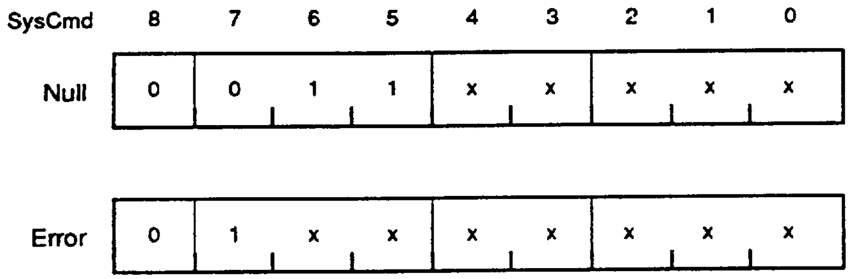
Various combinations are available depending on the lower bits of the address and data sizes. Details are described in a later chapter.

[Write Error]

The write error command is not generated by the VR4200. The BRAC ignores this command, and does not issue write commands to the memory. This is the same as the normal writing operation of the VR4200 interface. In other words, accessing is ignored.

4.4.4 Other request packets

Fig. 11 shows request packets ignored by the BRAC.



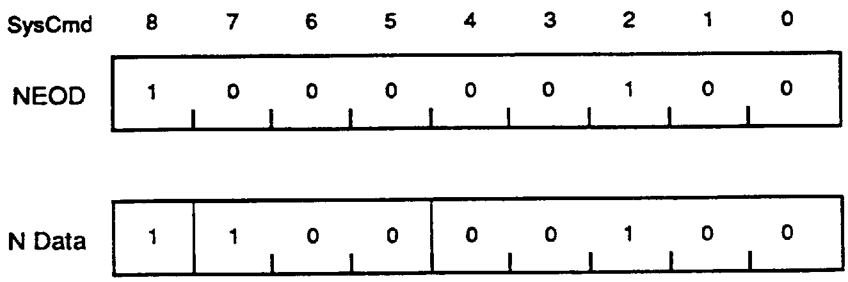
SysCmd	8	7	6	5	4	3	2	1	0
Null	0	0	1	1	x	x	x	x	x
Error	0	1	x	x	x	x	x	x	x

Fig. 11. Request Packets Ignored by BRAC

When the BRAC receives the request packets shown in **Fig. 11**, it will not operate until it receives the next valid request packets.

4.4.5 Data packets returned by BRAC

Fig. 12 shows read response packets returned by the BRAC.



SysCmd	8	7	6	5	4	3	2	1	0
NEOD	1	0	0	0	0	0	1	0	0
N Data	1	1	0	0	0	0	1	0	0

Fig. 12. Data Packets Returned by BRAC

The BRAC returns two types of data packets as described below.

[NEOD]

The last data element, response data, error-free data, and data checking.
Clean exclusive

[NData]

Other than the last data element, response data, error-free data, and data checking.
Clean exclusive

The data packets returned by the BRAC are transferred at the fastest speed if they continue. In other words, data is transferred continuously from the beginning to the end.

4.4.6 Data packets received by BRAC

Fig. 13 shows packets which are recognized as write data packets by the BRAC.

SysCmd	8	7	6	5	4	3	2	1	0
DATA	1	x	x	x	x	x	x	x	x

Fig. 13. Data Packets Received by BRAC

The BRAC acknowledge all valid data packets following the write command as write data. Notice that the contents of the packets are written sequentially into the write buffer automatically without being judged by the BRAC. The most significant bit of the data packets must be "1" for the SysCmd bus.

Data packets that satisfy these conditions are received after the write request and all of these data packets are written in the write buffer. Even if the number of data packets received differs from the transfer size of the write request, they are all received.

If the size of data to be written exceeds the capacity of the write buffer in the BRAC, some data in the buffer will be lost due to the FIFO mechanism. Therefore, do not write data that exceeds the write buffer capacity. The BRAC transmits Rambus packets according to the size identifier in the write request regardless of the number of data packets received.

4.4.7 Data location on SysAD bus in partial read/write

The BRAC operates according to the Little Endian ordering. The Rambus memory transfers at least 8 bytes (9 bits x 8) in one operation. Transfer begins with the lowest address in the Rambus memory space. The BRAC receives these data, stores them in a width of 64 bits from the LSB side byte by byte, and converts them to 64-bit width data. **Fig. 14** shows the addresses in the Rambus register space.

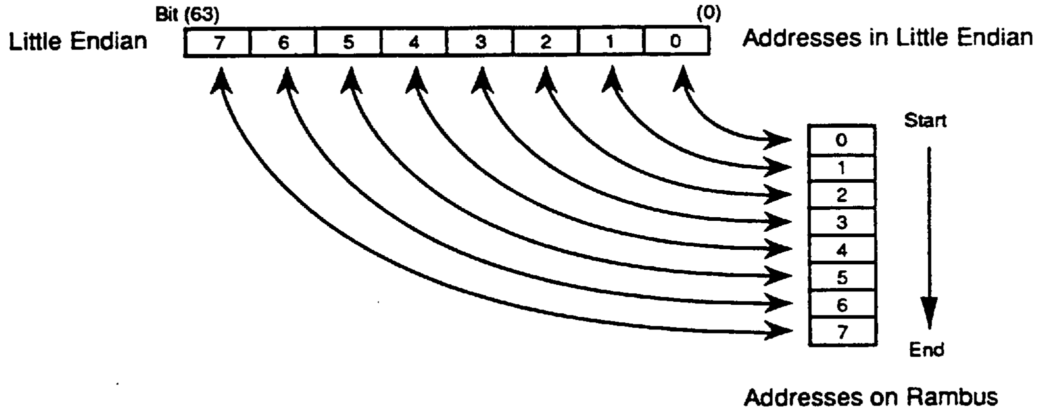


Fig. 14. Relation between Rambus Register Space and Addresses

Table 6 shows the relation between the number of transfer addresses/bytes and valid data lines on the SysAD bus.

Table 6. Data Transfer Address, Number of Bytes, and Location of Valid Data

	Number of Bytes							
Address	1	2	3	4	5	6	7	8
0	7: 0	15: 0	23: 0	31: 0	39: 0	47: 0	55: 0	63: 0
1	15: 8	23: 8	31: 8	39: 8	47: 8	55: 8	63: 8	63: 8
2	23:16	31:16	39:16	47:16	55:16	63:16	63:16	63:16
3	31:24	39:24	47:24	55:24	63:24	63:24	63:24	63:24
4	39:32	47:32	55:32	63:32	63:32	63:32	63:32	63:32
5	47:40	55:40	63:40	63:40	63:40	63:40	63:40	63:40
6	55:48	63:48	63:48	63:48	63:48	63:48	63:48	63:48
7	63:56	63:56	63:56	63:56	63:56	63:56	63:56	63:56

4.4.8 Data packets and transfer patterns

The BRAC data transfer patterns must be as follows:

[VR4200 → BRAC]

The data packets in a write command from the VR4200 accept all types of data patterns. The fastest pattern is "ADDDD", which is also valid.

(However, the VR4200 can only transmit "ADDxDDx" or "ADxxDxx".)

[BRAC → VR4200]

The data packet responding to the VR4200 read command can only be transmitted in the transfer pattern "D", which is the fastest pattern.

4.5 Rambus Interface

The Rambus transactions handled by the BRAC are as follows:

- Memory/Register read Transaction and Return OK
- Memory/Register read Transaction and Return Nack
- Memory/Register write Transaction and Return OK
- Memory/Register write Transaction and Return Nack

4.6 VR4200 and Rambus Request

Fig. 15 shows the physical memory space when the BRAC is used to configure the VR4200 system. In the figure, BRAC is set to make responses between addresses 0x0000 0000 and 0x4000 0000 by the internal register. The figure also shows that the RDRAM is actually mapped at only one part. The BRAC supports only 33-bit addresses and ignores the bits [35:33].

In this case, the memory space responds to the following accessing.

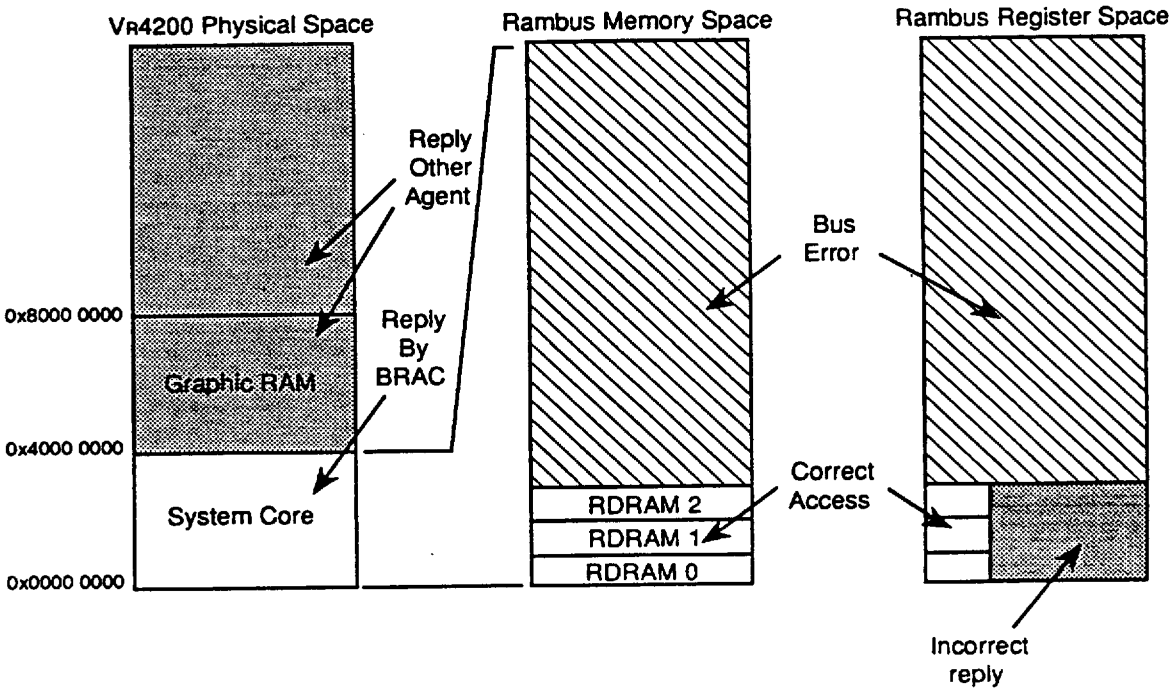


Fig. 15. Address Spaces and Bridge Responses

- **Reply-Other-Agent**

This area is reserved for mapping external agents except the BRAC. The BRAC cannot reply to this area.

Set the BRAC so that other external agents can reply to this space. The system will hang-up if there are no external agents when the processor issued read requests to this space. If an external agent is present, the agent issues a response.

If the processor issues write requests to this space, the requested external agent, if present, should operate according to the request.

The request is ignored if no agents are present.

- **Correct-Access**

  This area is provided for mapping the BRAC. The BRAC can reply to requests issued to this area.

  When the Rambus memory/register space is accessed, the BRAC returns the correct value. In this case, ACK response was returned when the Rambus space was accessed.

- **Incorrect-Reply**

  When the Rambus register space was accessed and the slave device was mapped in the address, the slave did not return the correct value.

  Because the Rambus slave device returns an ACK response, a response including incorrect data is returned to the VR4200.

- **Bus-Error**

  When the BRAC acknowledged a VR4200 request and issued the same request to an RDRAM, neither ACK nor NACK was returned.

  When the VR4200 issues a read request, the BRAC returns response data with errors. When the VR4200 issues a write request, the BRAC outputs a signal to the outside indicating that an error has occurred.

5 REGISTERS

The following describes the groups of registers and addresses controlling the operations of BRAC and displaying their operation statuses.

5.1 Internal Register Address

The BRAC has several control registers. These registers are mapped in the fixed addresses of the VR4200. These addresses are reserved and cannot be used for other purposes. In addition, the internal registers of the BRAC can only be accessed by the 64-bit single access method. Normal operations cannot be guaranteed when other access methods (below 64 bits and block transfer access) are used.

The BRAC supports the 33-bit address space.

These modes are described separately below.

5.1.1 Address space

Table 7 shows the address map of the BRAC register for the 33-bit address. The operations of each register will be described later.

By this setting, this reserved space may be overlapped with the Rambus window of the BRAC. In this case, the internal register will be accessed first. In other words, the addresses from 0x1 FFFF FF00 to 0x1 FFFF FFFF are fixed for the BRAC internal register.

Internal register address decoding is performed for the address bits [32:0]. The address bits [35:33] are ignored. Consequently this address will also be ignored when windows to the Rambus space are selected. All addresses for the Rambus space [35:33] will be masked and "0" will be output.

Internal registers are accessed for 64-bit single read/write commands only. Read response/write operations are not guaranteed when other types are accessed.

Table 7. BRAC Internal Register Address Map (33-Bit Mode)

Abbreviation	Name	Address	Initial Value
RABR	Address base page	0x1 FFFF FF00-0x1 FFFF FF07	0x 0000 0000 0000 0000
RAMR	Address mask	0x1 FFFF FF08-0x1 FFFF FF0F	0x 0000 0000 0000 0000
MODE	Mode delay	0x1 FFFF FF10-0x1 FFFF FF17	0x 0000 0000 0000 0000
CONF	Configuration	0x1 FFFF FF18-0x1 FFFF FF1F	0x 0000 0000 0000 0000
ERROR	Error address register	0x1 FFFF FF20-0x1 FFFF FF27	Undefined
RSRV	Reserved space	0x1 FFFF FF28-0x1 FFFF FFFF	Undefined

5.2 SysAD Bus Control Registers

Group of registers controlling the BRAC SysAD bus interface.

- **Rambus address base page register (RABR)**

This is the Base register that is used to specify the address for converting from the SysAD bus request to the Rambus request. **Fig. 16** shows the allocation of this register. Only the bits [35:16] can be used for this register.

The bits [35:33] of this register have no meaning. Operations however will not be guaranteed when other than "0" is set. The "RFU" bit is undefined and writing will be ignored.

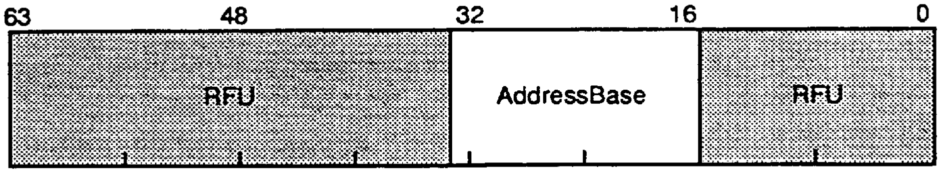


Fig 16. Address Base Register

- **Rambus Address Mask Register (RAMR)**

This is the mask which is used to decide the effective space of the Rambus address base page register. The location of this register is shown in **Fig. 17.** The bits [35:16] can be used for this register. If a bit is assigned as "0", the corresponding bit of the address will be masked. The addresses corresponding to the bits [15:0] are always masked.

The bits [35:33] of this register have no meaning. Operations however will not be guaranteed when other than "0" is set. This register decides the Rambus window and does not affect addresses output to the Rambus. The "RFU" bit is undefined and writing will be ignored.

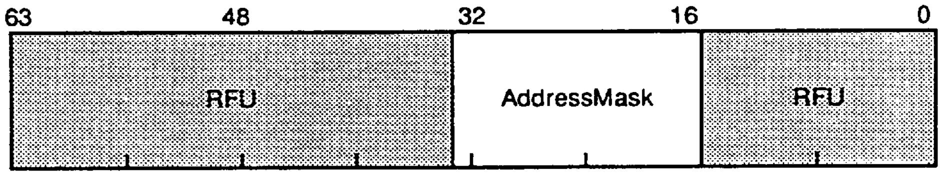


Fig 17. Address Mask Register

When the value obtained by masking the execution address with RAMR and that obtained by masking RARB with RAMR are the same, the BRAC converts the request to Rambus. With this method, the 2n (16<n<33) space can be set.

5.3 Mode Registers

The mode registers are used to specify the parameters for transmitting Rambus packets.

The values are the same as the Rambus technical description mode registers. **Fig. 18** shows the bit patterns.

63							56		48		40			32
Rambus_Reset	AE	SIn	SOut	RacReset	StopT	StopR	Pwr_Down	sck_dly [3:0]	ra_dly [3:0]	RambusRetry Count [3:0]	RambusMax Retry Count [3:0]	CCtlEn	CCtLd	C [5:0]

[Out Mode]

31	24		16		8		0
AckWin Mask	AckWin Delay	rcv_Adj	rcv_dly	AckWin WrMask	AckWin WrDelay	snd_adj	snd_dly

Fig. 18. Mode/Delay Registers

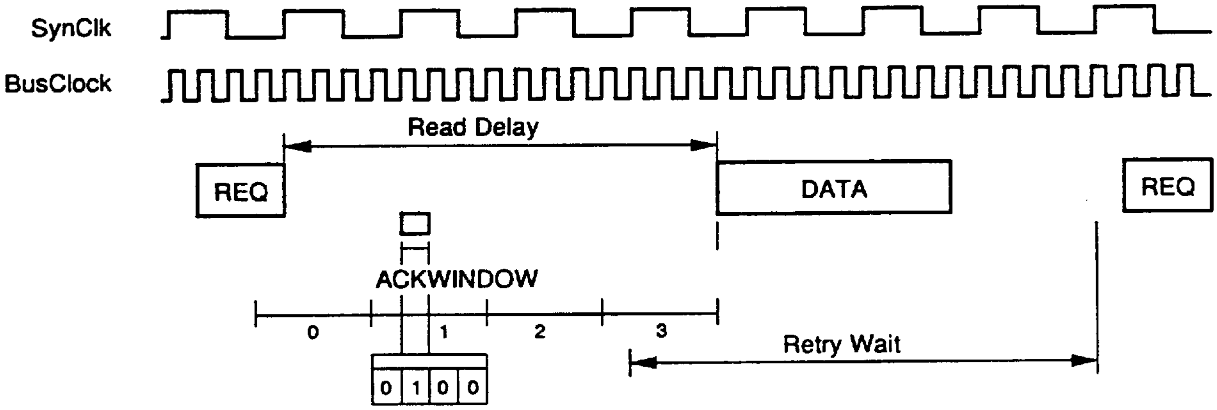


Fig. 19. Mode/Delay Parameters

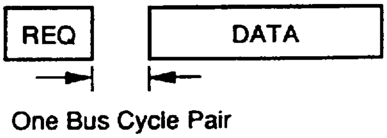


Fig. 20. Write Delay Parameters

rambus reset The Rambus can be reset by writing "1" and setting the BusEnable of Rambus to the "L" level. By setting this register to "1" for a fixed period of time and then setting it to "0", the RDRAM can be reset.

AE Address window active. When this register is set to "1", the Rambus window of the BRAC turns ON and accessing starts for Rambus. When it is set to "0", the Rambus space will not be accessed.

SIn Receives the external pin SIn value.

SOut Affects the external pin SOut value.

RacReset Resets the RAC (Rambus control unit) inside the BRAC. When the power for the BRAC is turned on, or the BRAC is recovered after power down of the Rambus interface of the BRAC, it is necessary to maintain the "1" written for a certain time before performing the first access. In normal use, "0" must be written.

StopT By writing "1", the transmission logic clock of the Rambus interface PLL can be stopped. When accessing again after the clock is stopped, take note of the re-oscillation of the clock.

StomR By writing "1", the reception logic clock of the Rambus interface PLL can be stopped. Exercise due care when accessing after the clock is stopped.

Pwr_Down Power down. When "1" is written, the Rambus interface will power down completely. To recover, the "Rambus" interface must be initialized by setting Pwr_Down to "0" and RacReset to "1".

sok_dly The number of SynClk cycles from the end of the request packet until the assertion of SOK.

ra_dly The number of Synclks from the end of the read request packet to the start of serial packet transfer that specifies the address for the read request.

RambusMaxRetryCount The number of re-request issues when the NACK packet is received from the Rambus.

RambusRetryCount The number of SynClk cycles to the start of retry after transmitting the request packet to the Rambus channel.

CCtlEn Connected to the CCtlEn of the RAC.

CCtlLd Connected to the CCtlLd of the RAC.

CCtlI Connected to the CCTlI of the RAC.

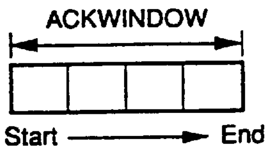


Fig. 21. ACKWINDOW Parameter

AckWinMask Sets the timing for determining the ACK bits of the Rambus window during read request. **Fig. 21** shows the window. ACK is determined at the timing at which "1" was written. When writing several "1"s, several ACK WINDOWs can be used.

AckWinDelay The number of SynClk cycles between the SynClk cycle containing the end of the request packet and the SynClk cycle for searching the ACK packet during read requests.

rcv_adj The BDSel, BCSel, and BESel values during read requests. Timings can be adjusted by Rambus bus cycle pairs using this.

rcv dly The number of SynClk cycles from the end of the request packet to the beginning of the read data. By using this together with rcv adj, the timings shown in **Fig. 20** can be set for more than one bus cycle pair.

AckWrWinMask The number of Syn-Clk cycles from the end of the request packet to the ACK packet search during write requests. The timing used for determining the ACK bit of the Rambus window is set during write requests. **Fig. 21** shows the window. ACK is determined using the timing at which "1" was written. When writing several "1"s, several ACK WINDOWs can be used.

AckWrWinDelay The number of Syn-Clk cycles from the end of the request packet to the ACK packet search during write requests.

snd_adj BDSel, BCSel, and BESel values during write requests. Timings can be adjusted by Rambus bus cycle pairs using this.

snd_dly Delay time measured by SynClks between the end of the request packet to the beginning of the write data. By using this together with snd_adj, timings can be adjusted in units of one bus cycle pair.

5.4 Configuration Register

This is the register which is used to preserve the inherent parameters of the BRAC. See **Fig. 22** below.

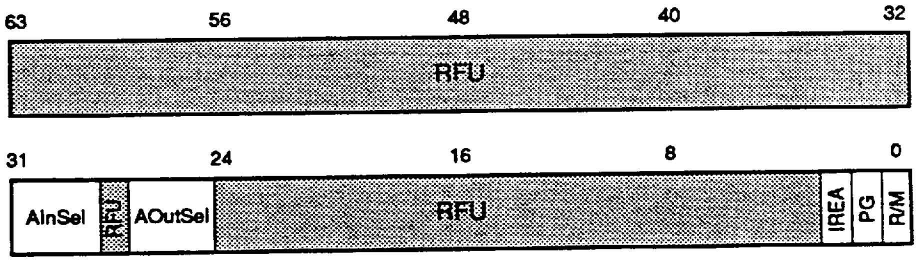


Fig. 22. Configuration Register

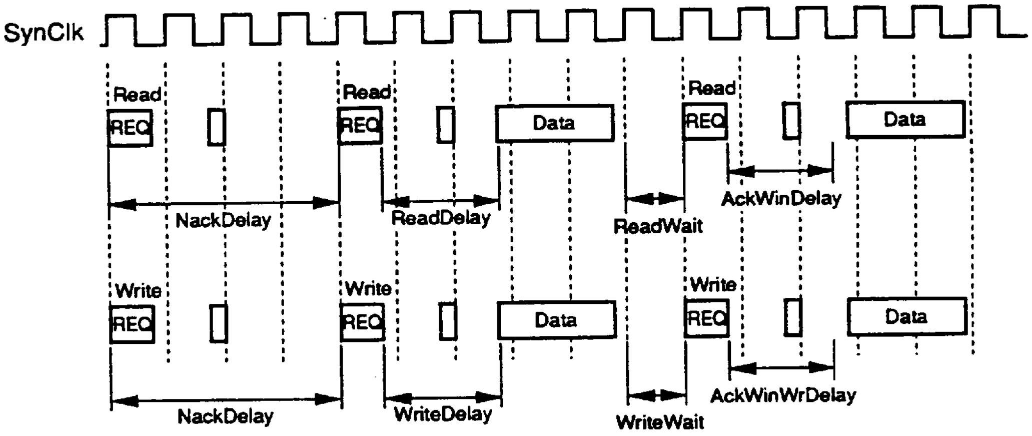


Fig. 23. NackDelay/ReadWait/WriteWait Parameters

- **R/M**

  When this bit is set to "0", the VR4200 request is converted to the request for the Rambus register space.

  When this bit is set to "1", the VR4200 request is converted to the request for the Rambus memory space.

- **PG Parity Generation**

  0: All parities are generated inside the BRAC.

  1: When accessing the Rambus memory space, the Rambus parity will be used.

- **IREA**

  Initialization register. When "1" is written, SysAD addresses are used as is for Rambus devices. When "0" is written, the Rambus address bits [33:21] are masked with "0".

- **AInSel**

  Specifies the input sample timing of the AsynClk space.

- **AOutSel**

  Specifies the output valid data window timing of the ASynClk space.

- **RFU**

  RFU bits are fixed to 0 and writing will be ignored.

6 OUTLINE OF OPERATION TIMINGS

6.1 Block Read Responses (Prefetch Buffer Hits)

Fig. 24 shows an example of the block read response when the prefetch buffer hits.

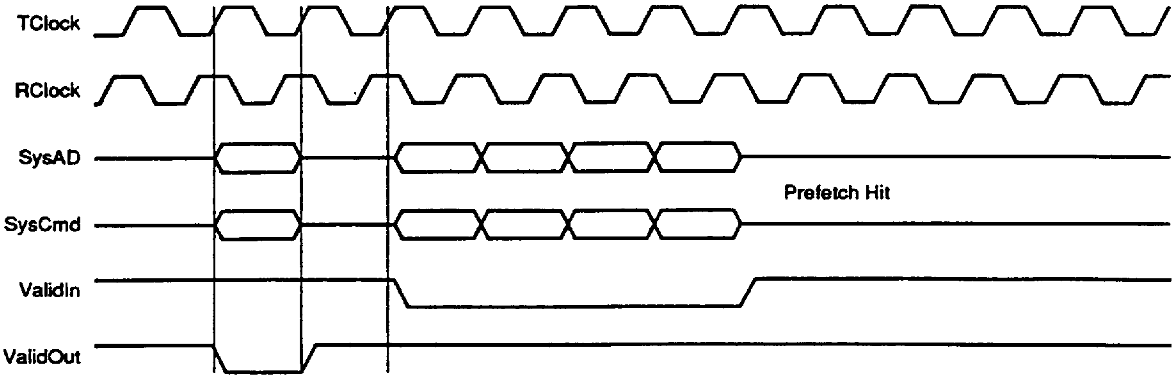


Fig. 24. Block Read Response
(Prefetch Buffer Hit)

- "RdRdy" is negated one clock after the VR4200 SysAD bus read request is acknowledged.
- When the VR4200 SysAD bus read request is acknowledged, prefetch buffer is judged after imposing one TClock clock.
- Data is output continuously when the prefetch buffer is hit.

6.2 Block Read Responses (Prefetch Buffer Misses)

Fig. 25 shows an example of the block read response when the prefetch buffer is not hit correctly.

The BRAC converts the VR4200 block read request to the Rambus non-sequential read request that reads the address of the request first.

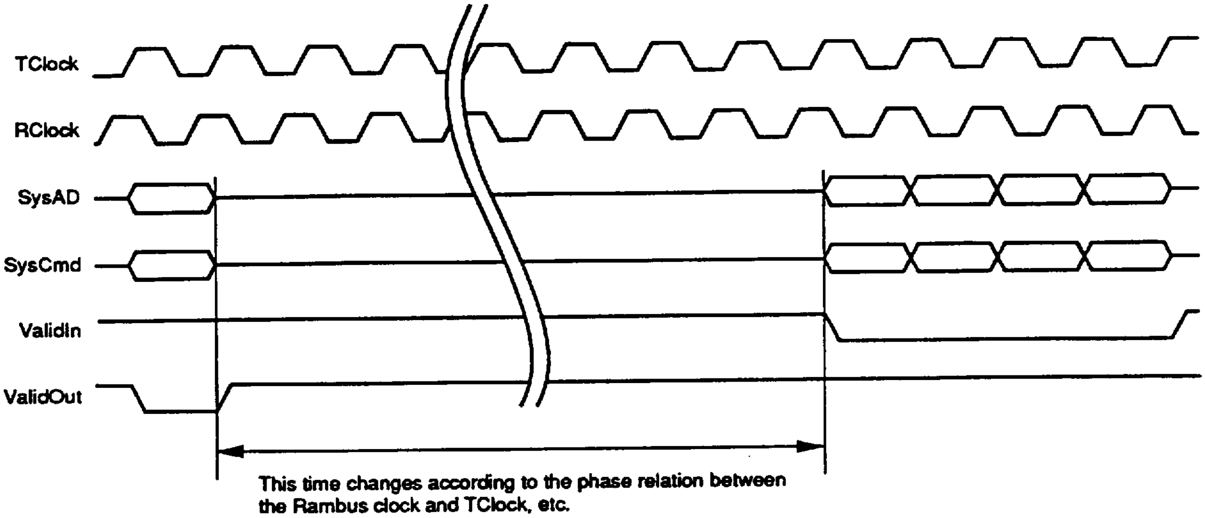

TClock
RClock
SysAD
SysCmd
ValidIn
ValidOut
This time changes according to the phase relation between the Rambus clock and TClock, etc.

Fig. 25. Block Read Responses (Prefetch Buffer Miss)

- "RdRdy" is negated one clock after the VR4200 SysAD bus read request is acknowledged.
- When the VR4200 SysAD bus read request is acknowledged, prefetch buffer is judged after imposing one TClock clock.
- The RambusDRAM is accessed, and the data is read and then output when the prefetch buffer is not hit correctly.

6.3 Double-word, Word, Partial Word Read

Fig. 26 shows the response packet for the partial read request.

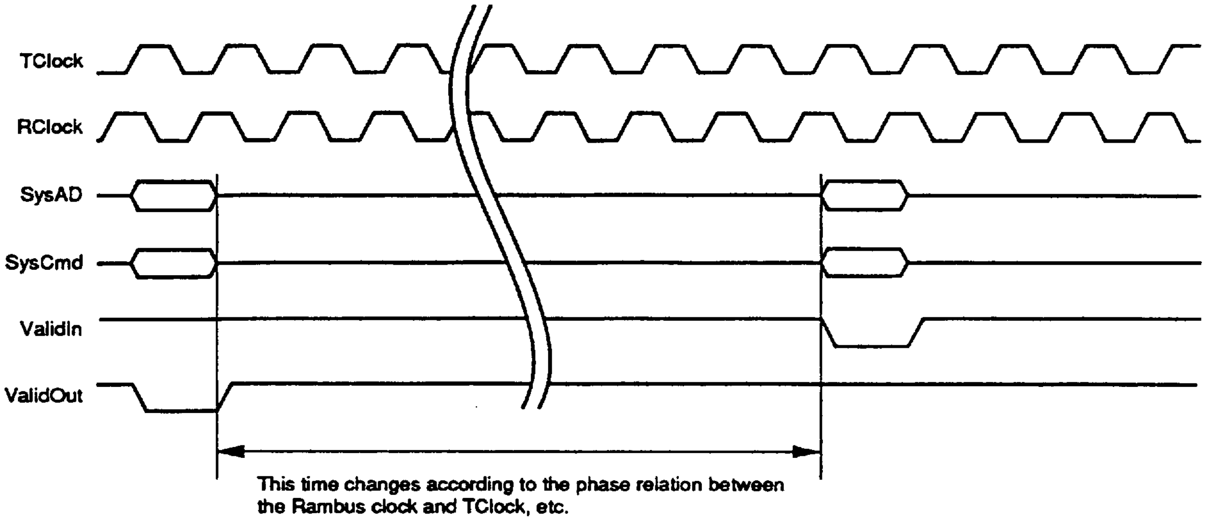

TClock
RClock
SysAD
SysCmd
ValidIn
ValidOut
This time changes according to the phase relation between the Rambus clock and TClock, etc.

Fig. 26. Partial Read Responses (Prefetch Buffer Miss and Hit)

6.4 Read Requests/Error Termination

When the Rambus does not return the acknowledge to the "ACK" window, the VR4200 read request ends returning the error, as shown in **Fig. 27**.

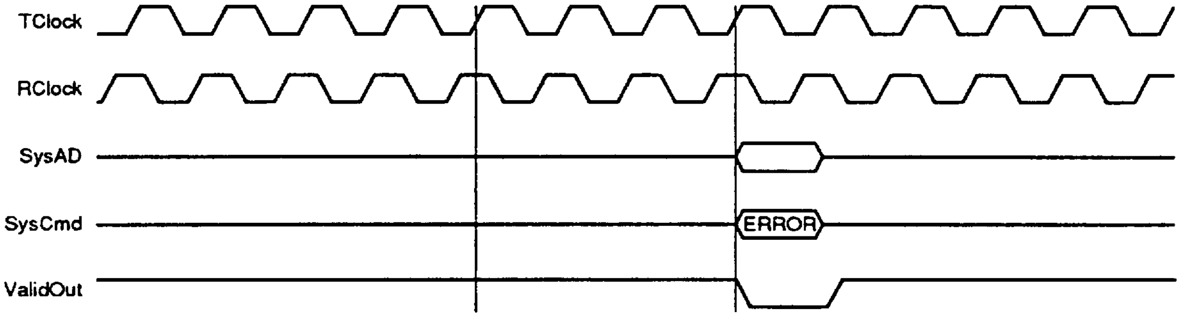


Fig. 27. Error Responses

6.5 Write Requests

Fig. 28 shows an example of block write request processing timing using 8-word block transfer. During writing, the write request to Rambus devices is issued after the first data packet from VR4200 is received.

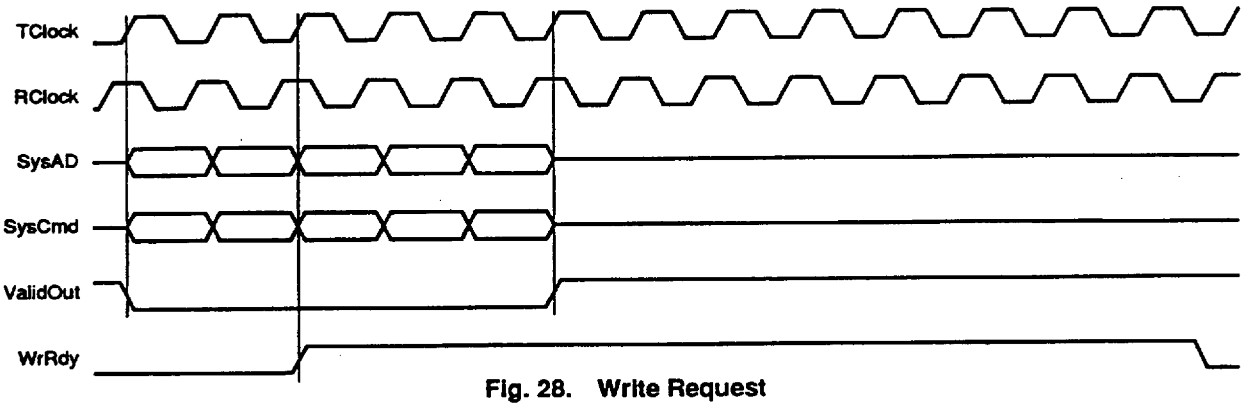


Fig. 28. Write Request

- The write request has no response, and therefore the VR4200 does not receive errors corresponding to the write requests.
- The WrRdy signals is negated one cycle after the VR4200 receives the write request commands.
- It ignores and will not accept the next SysAD request until the WrRdy signal is asserted.
- The WrRdy signal is asserted when it has been written in the RDRAM memory.
- A request to Rambus is generated when the write request command and first data is received.
- The BRAC does not permit any kind of packets between write commands and the data. In other words, the SysAD bus master must not generate bus release acknowledges for the ExtReq before the write command is completed.

6.6 Write Requests/Error Termination

If RDRAM does not return an acknowledge response during the "ACK" window, it enables an "error" signal which indicates that an error has occurred, and terminates the operation in the same way as normal write operation.

7 RESET

BRAC must perform the reset sequence using the BRAC internal register and RDRAM register. By transmitting the command from SysAD bus, the sequence can be written in the register. **Table 8** shows the sequence when the BRAC is connected to the SysAD bus of the VR4200 and one NEC 18MRDRAM is connected to Rambus interface of the BRAC.

This example shows the sequence for when RDRAMs are mapped at 0x20000000 to 0x201fffff of the VR4200. The address of the RDRAM register should be set accordingly.

[Procedures and Operations]

- WAIT ("TIME"):
  Function which waits for clocks indicated by time. The unit is TTCY K (TClock cycle).

- WRITE ("Address", "value", "size"):
  Function which transmits a single write request to the SysAD bus. The size is specified in byte units.

Table 8. BRAC Initialization Sequence 1/2

```
void BRAC-Initialize () {
   WAIT (20);                                  /* Certain power supply voltage and clocks must be applied  */
                                               /* continuously to BRAC.                                     */
   WRITE ( MODE , 0x88101f00f213f411 ,8);      /* Starts resetting the RDRAM by writing 1 in the "Rambus    */
                                               /* Reset" of BRAC.                                           */
                                               /* As the RDRAM REFERENCE MANUAL (Version 1.0)               */
                                               /* indicates that the reset time is 544 ns at the minimum,   */
                                               /* maintain the "Rambus Reset" at "1" above this time.        */
   WAIT (3);                                   /* Starts resetting RAC by writing "1" in "RAC Reset".        */
   WRITE ( RABR , 0x0000000020000000 ,8);      /* Sets the address mapping RDRAM to "0x2000000" by          */
   >WAIT (3);                                  /* writing "0x2000000" in the RABR register of BRAC.          */
   WRITE ( RAMR , 0x00000001001fffff ,8);      /* Sets so that all banks of 18MRDRAM will be used by         */
   WAIT (3);                                   /* writing "0x1001fffff" in the RAMR register of BRAC.        */
```

Table 8. BRAC Initialization Sequence 2/2

```
    WRITE ( CONF , 0x00200000a0000000 ,8);       /* Sets the following using the RDRAM CONF register and  */
                                                 /* prepares to write the settings in the RDRAM register  */
                                                 /* space.                                                 */
                                                 /* Address Window=Enable,                                 */
                                                 /* Reg/Mem=Reg,                                           */
    WAIT (3);                                    /* IREA=0 (Addr [36:21]=0)                                */
    WRITE ( MODE , 0x50102200f314f421 ,8);       /* Sets the following using the MODE register of BRAC.   */
                                                 /* Address window active                                  */
                                                 /* Starts resetting RDRAM by writing "1" in "Rambus Reset". */
                                                 /* Ends the RAC resetting by writing "0" in "RAC Reset".  */
                                                 /* The following are examples of these settings.          */
                                                 /* AckWinDelay=3, AckWinMask="1111"                       */
                                                 /* rcv dly=4, rcv adj="0001"                              */
                                                 /* AckWinWrDelay=4,                                       */
                                                 /* AckWinWrDelay="1111",                                  */
    WAIT (6);                                    /* snd dly=1, snd adj="0010"                              */
    WRITE ( RDRAM#0 Mode , 0x0000000200000000 ,4);  /*                                                     */
                                                 /* Writes "1" in DE bit of the Mode register of RDRAM. This */
    WAIT (10);                                   /* will enable RDRAM to be used.                          */
    WRITE ( RDRAM#0 DeviceID , 0x0004000000000000 ,4);  /*                                                 */
                                                 /* Sets the RDRAM device ID register and sets the RDRAM  */
    WAIT (10);                                   /* address to "0x2000000".                                */
    WRITE ( CONF , 0x00201300a0000007 ,8);             /*                                                  */
                                                 /* Sets the following using the BRAC CONF register.       */
                                                 /* Address Window=Enable,                                 */
    WAIT (3);                                    /* Reg/Mem=Mem, IREA=1                                    */
}
```

8 ELECTRICAL SPECIFICATIONS

(These specifications are target values and may be changed without prior notice.)

8.1 Absolute Maximum Ratings

Table 9. Absolute Maximum Ratings

Parameter	Symbol	Ratings	Unit
Power supply voltage	VDD	–0.3 to +3.6	V
Input voltage	VI	–0.3 to VDD+0.3	V
Output voltage	VO	–0.3 to VDD+0.3	V
Operating ambient temperature (case surface)	T_{copt}	0 to +70	°C
Storage temperature	T_{stg}	–65 to +150	°C

8.2 DC Characteristics

The VR4000 interface of BRAC operates at 3.3 V and in CMOS level. It is not interchangeable with the 5 V interface. **Table 10** shows the input/output specifications.

Table 10. Interface Pins DC Characteristics

(Tcopt=0 to +85 °C)

Parameter	Symbol	Min	Max	Unit	Condition
CMOS L level input voltage	VCIL	–0.3	0.8	V	
CMOS H level input voltage	VCIH	2.3	VDD+0.3	V	
CMOS L level output voltage	VCOL	0.0	0.5	V	
CMOS H level output voltage	VCOH	2.6	VDD	V	
CMOS input leak current	ICLI		TBD	µA	
CMOS output leak current	ICLO		TBD	µA	
Rambus termination voltage	V_{Rterm}	2.2	2.7	V	
Rambus reference voltage	V_{Rref}	1.7	2.4	V	
Rambus H level input voltage	VRIH	VREF+0.2		V	
Rambus L level input voltage	VRIL		VREF–0.2	V	
Rambus H level output voltage	VROH	VREF+0.3		V	
Rambus L level output voltage	VROL		VREF–0.3	V	
Power voltage (digital section)	VDD	3.135	3.465	V	
Power voltage (analog section)	VDDA	VDD–0.1	VDD+0.1	V	
Power supply current	IDD		TBD	mA	f=250 MHz

8.3 VR4200 AC Timings

8.3.1 Clock timings

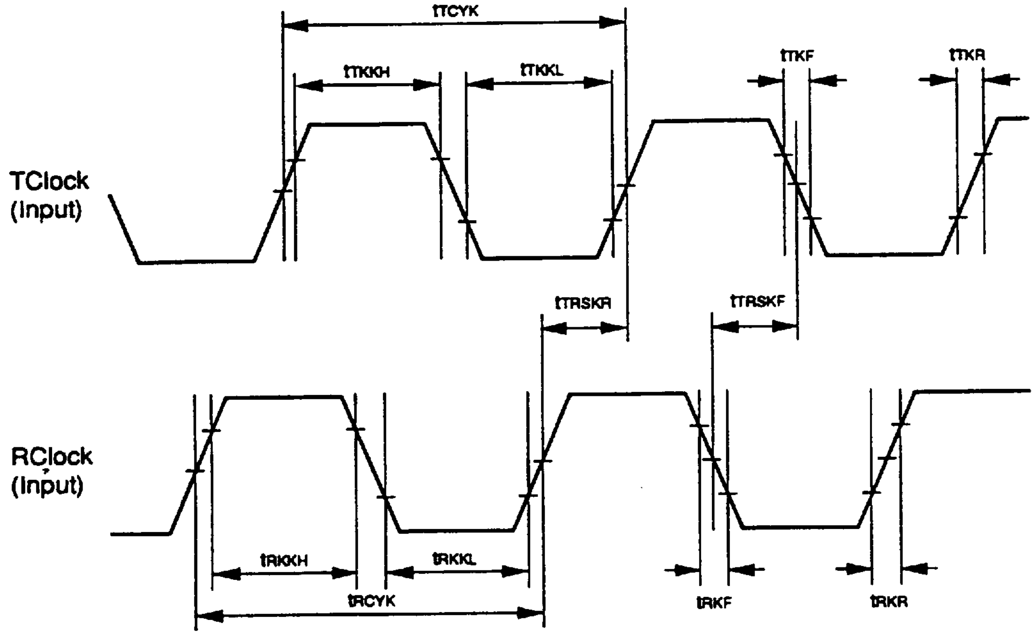


Fig. 29. Clock AC Timings

Table 11. Clock Timings

Parameter	Symbol	Min	Max	Unit
TClock cycle	t_{TCYK}	25.0		ns
TClock H level width	t_{TKKH}	12.0	13.0	ns
TClock L level width	t_{TKKL}	12.0	13.0	ns
TClock-RClock rising skew	t_{TRSKR}	t_{TCYK}/4-1	t_{TCYK}/4-1	ns
TClock-RClock falling skew	t_{TRSKF}	t_{TCYK}/4-1	t_{TCYK}/4-1	ns
RClock cycle	t_{RCYK}	25.0		ns
RClock H level width	t_{RKKH}	12.0	13.0	ns
RClock L level width	t_{RKKL}	12.0	13.0	ns

8.3.2 Reset timings

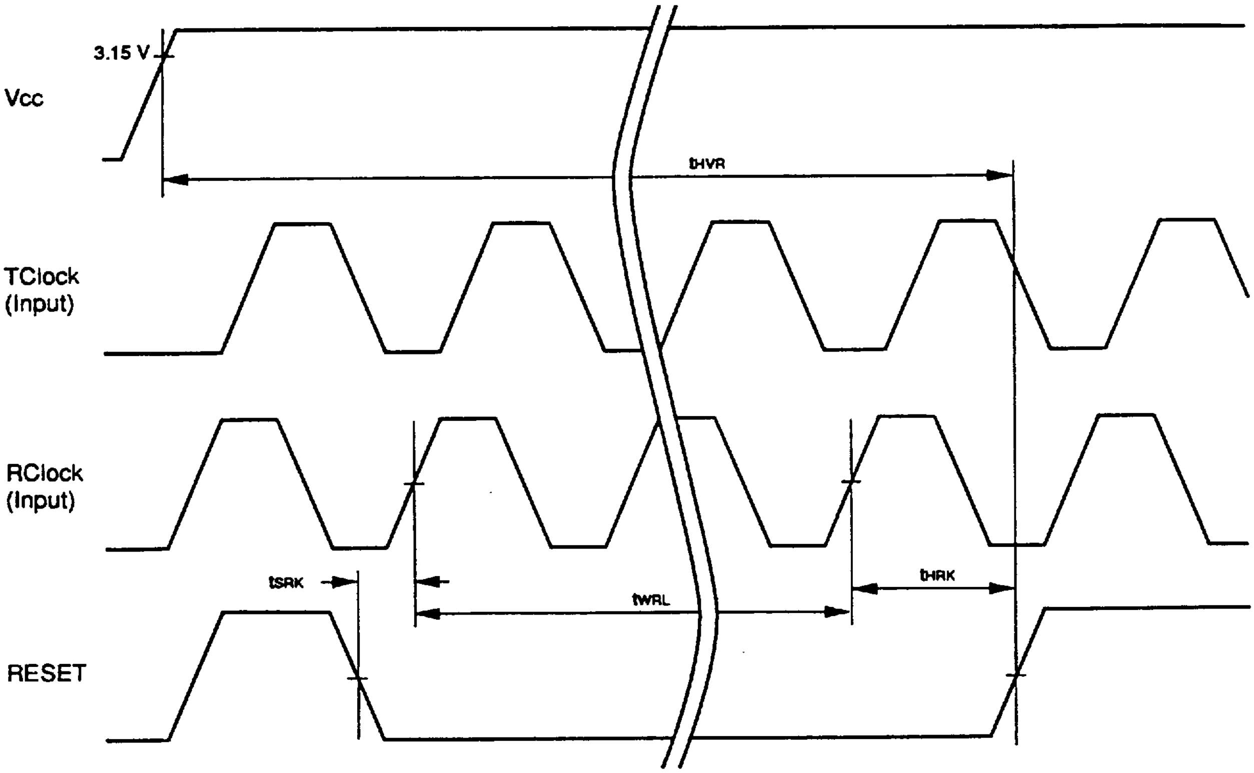


Fig. 30. Reset Timings

Table 12. Reset Timings

Parameter	Symbol	Min	Max	Unit
Reset input width (V_{DD})	t_{HVR}	100		t_{CYKR}
Reset input width (clock)	t_{WRL}	100		t_{CYKR}
Reset input setup time	t_{SRK}	20		ns
Reset input hold time	t_{HRK}	10		ns

8.3.3 Input setup hold time

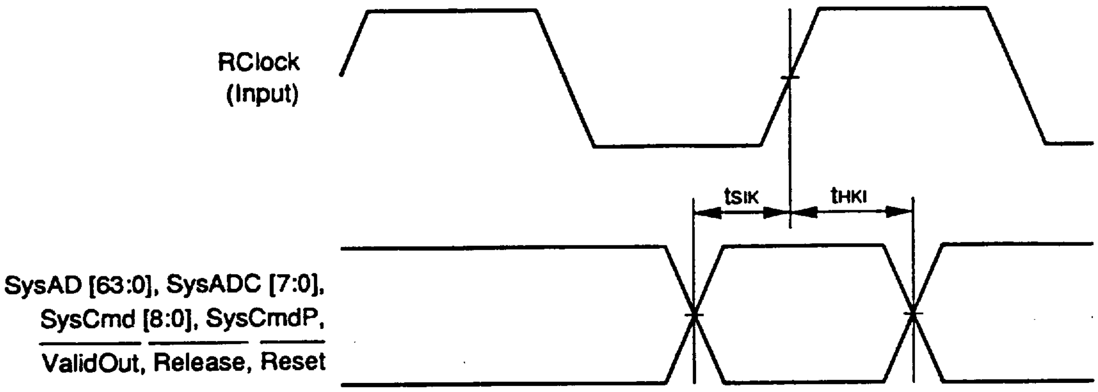


Fig. 31. Input AC Timing

Table 13. Input Timings

Parameter	Symbol	Min	Max	Unit
Input setup time	t_{SIK}	10		ns
Input hold time	t_{HKI}	5		ns

8.3.4 Output delay time

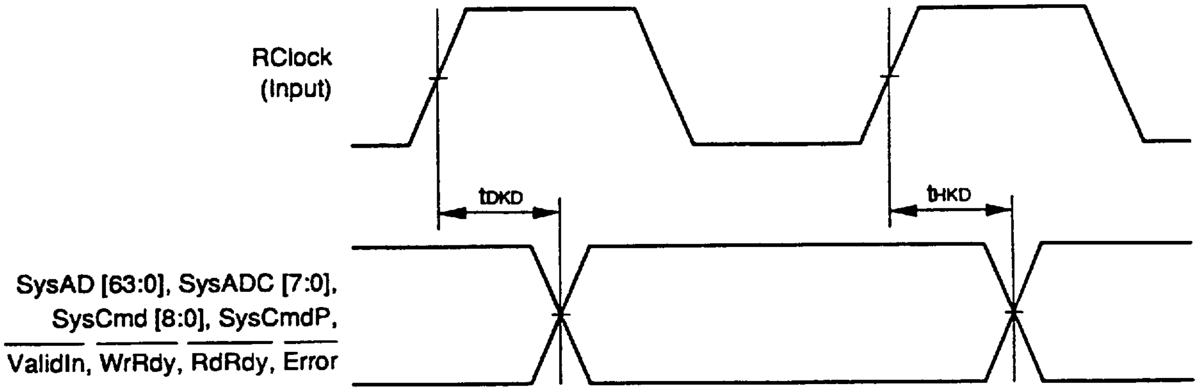

RClock
(Input)
t_{DKD}
t_{HKD}
SysAD [63:0], SysADC [7:0],
SysCmd [8:0], SysCmdP,
$\overline{\text{ValidIn}}$, $\overline{\text{WrRdy}}$, $\overline{\text{RdRdy}}$, $\overline{\text{Error}}$

Fig. 32. Output Delay Timings

Table 14. Output Delay Timings

Parameter	Symbol	Min	Max	Unit
Output active delay time	t_{DKD}	0	10	ns
Output inactive delay time	t_{HKD}	0	10	ns

8.3.5 Output floating delay

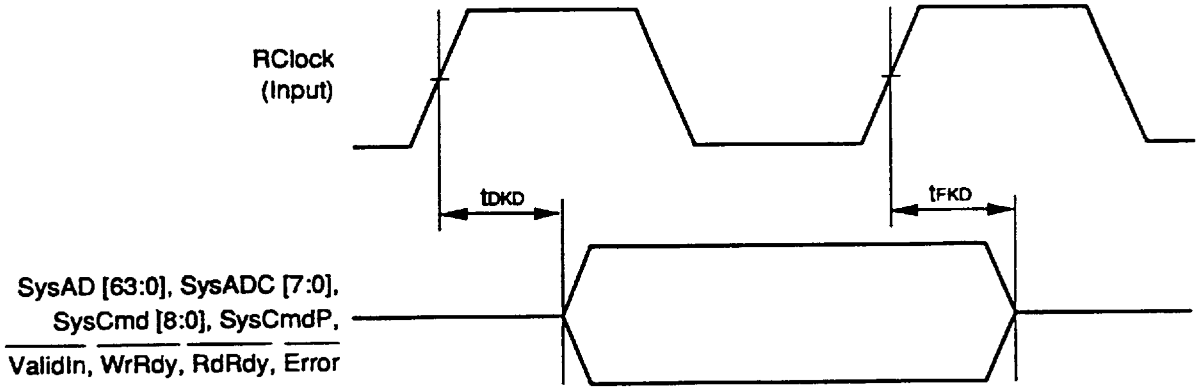

RClock
(Input)
t_{DKD}
t_{FKD}
SysAD [63:0], SysADC [7:0],
SysCmd [8:0], SysCmdP,
$\overline{\text{ValidIn}}$, $\overline{\text{WrRdy}}$, $\overline{\text{RdRdy}}$, $\overline{\text{Error}}$

Fig. 33. Output Floating Delay Timings

Table 15. Output Floating Delay Timings

Parameter	Symbol	Min	Max	Unit
Floating-to-active delay time	t_{DKD}	0	10	ns
Active-to-floating delay time	t_{FKD}	0	15	ns

8.4 Rambus AC Timings

8.4.1 Rise/Fall timings

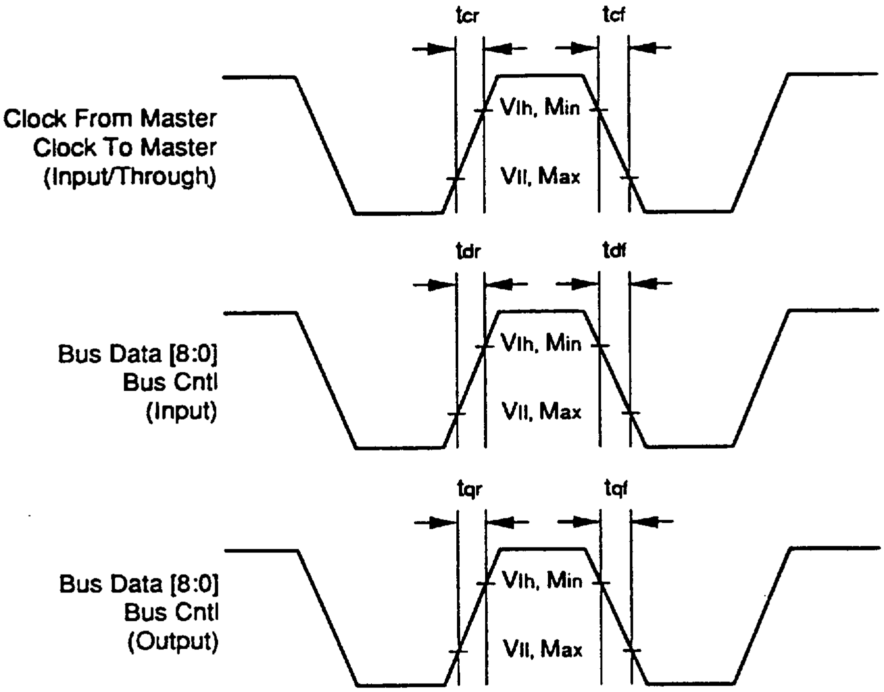


Fig. 34. Input/Output Signal Rise/Fall Timings

Table 16. Signal Rise/Fall Timings

Parameter	Symbol	Min	Max	Unit
Input/through clock rise time	tcr	0.3	0.7	ns
Input/through clock fall time	tcf	0.3	0.7	ns
Input signal rise time	tdr	0.3	0.7	ns
Input signal fall time	tdf	0.3	0.7	ns
Output signal rise time	tqr	0.4	0.6	ns
Output signal fall time	tqf	0.4	0.6	ns

8.4.2 Clock timings

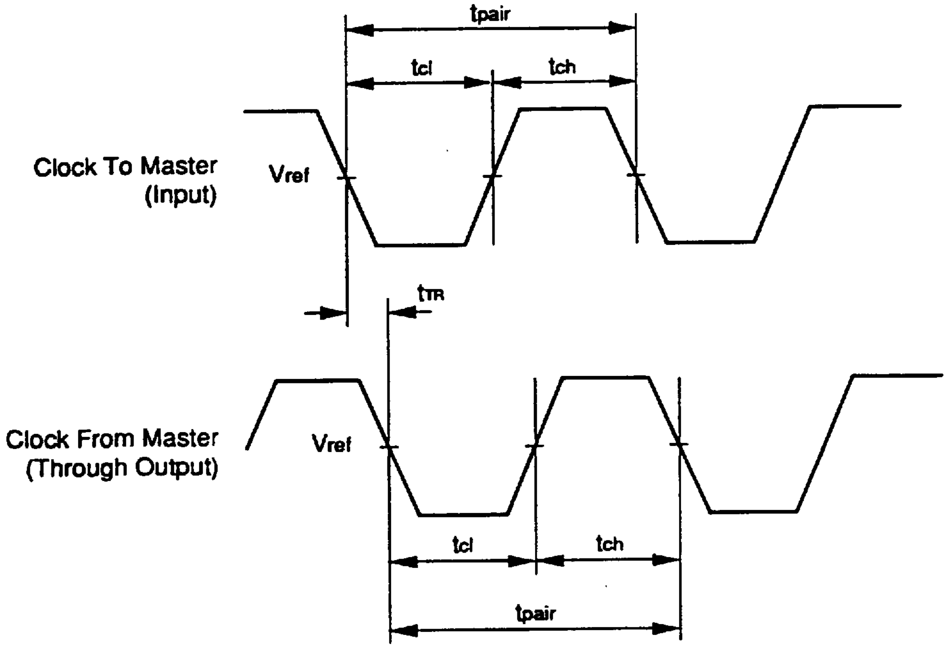


Fig. 35. Clock Timings

Table 17. Clock Input/Output Timings

Parameter	Symbol	Min	Max	Unit
Input clock cycle time	t_{pair}	4	5	ns
Input clock Hi time	t_{ch}	1.3	2.7	ns
Input clock Lo time	t_{cl}	1.3	2.7	ns
Input/output clock delay time	t_{TR}	0	0.3	ns
Input clock jitter	t_{ji}	0	50	ps
Input clock slew rate	t_{vl}	0	500	ps/s

8.4.3 Input timings

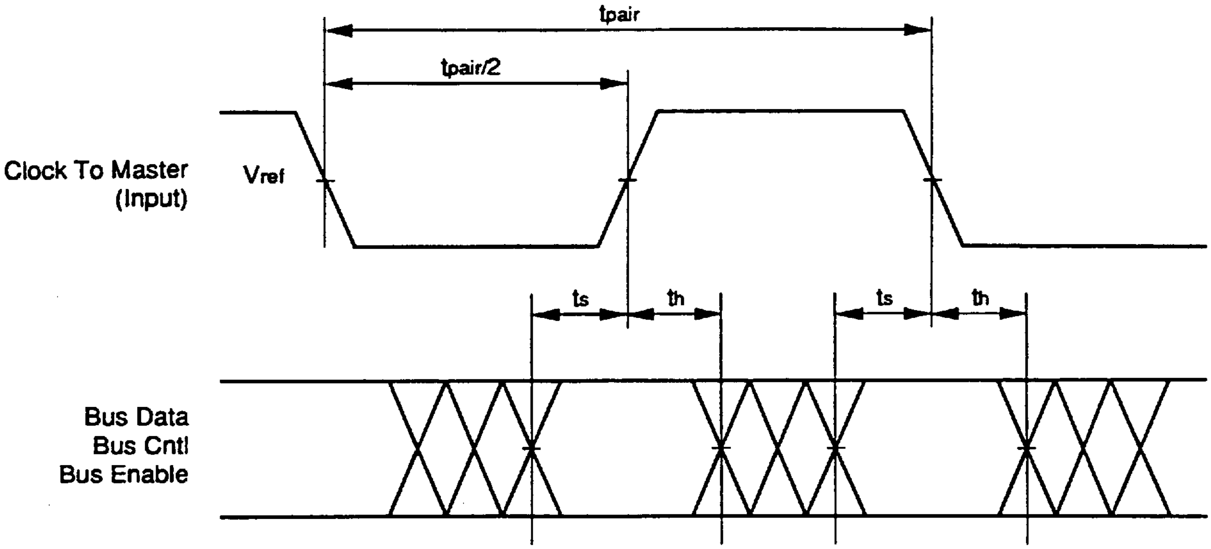


Fig. 36. Input Timings

Table 18. Input Timings

Parameter	Symbol	Min	Max	Unit
Input data setup time	t_s	0.3		ns
Input data hold time	t_h	0.3		ns

8.4.4 Output timings

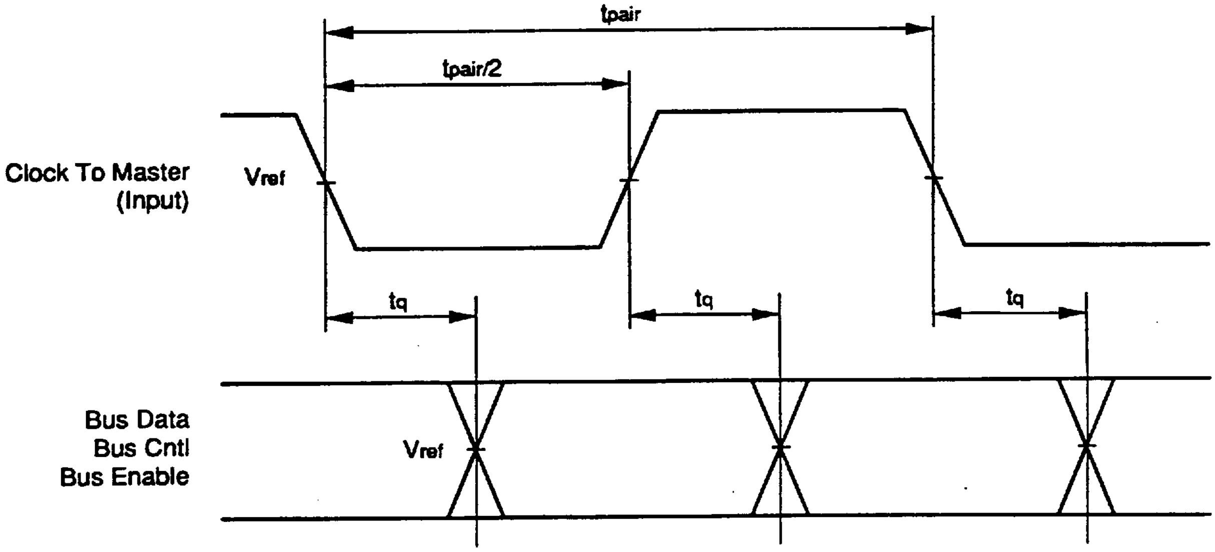


Fig. 37. Output Timings

Table 19. Output Floating Delay Timings

Parameter	Symbol	Min	Max	Unit
Output data delay time	t_q	$t_{pair}/4-0.3$	$t_{pair}/4+0.3$	ns